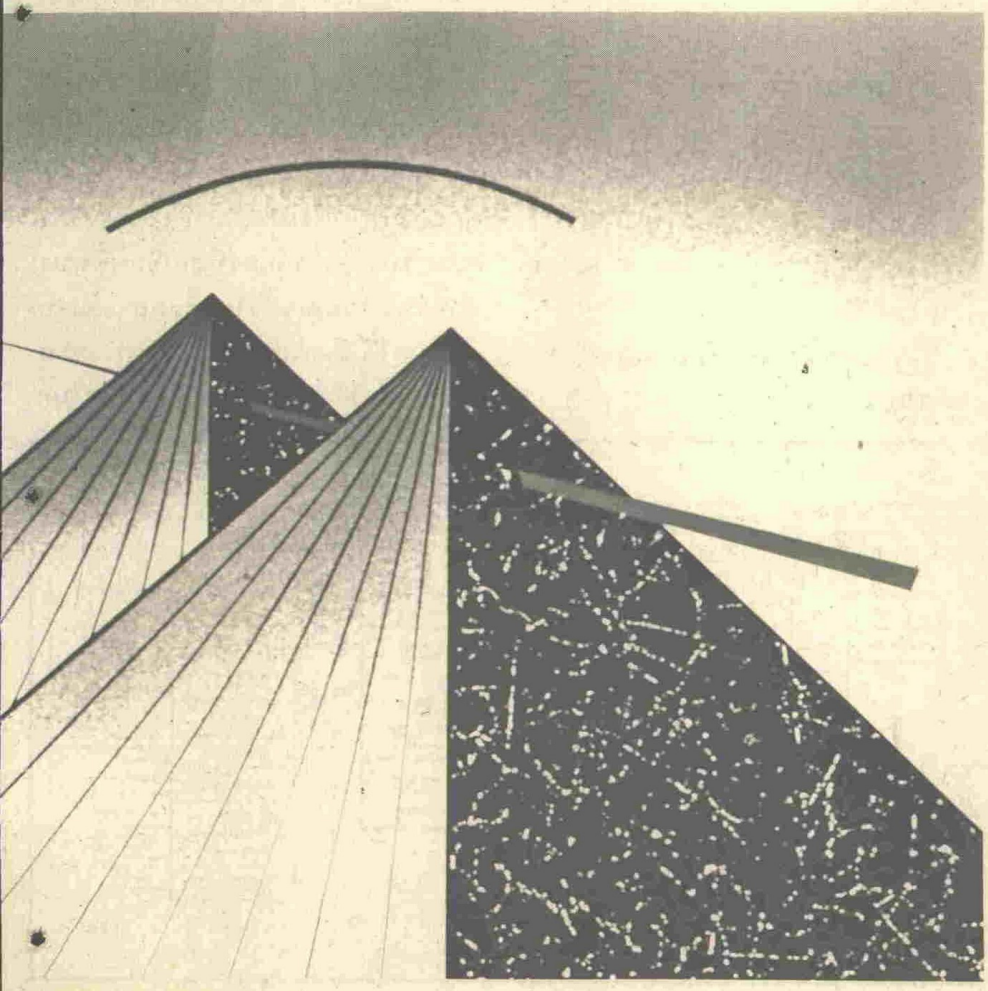
เอนก ศิรจริยาภิวัฒน์

ก้าวทันสมัยด้วยการรู้จัก 80486SX

แนะนำซีพียูใหม่ จากอินเทลที่มาเติมช่องว่าง ระหว่าง 80386 กับ 80486



เรื่องของซีพียูของค่ายอินเทลในตอนนี้มักจะปรากฏอยู่เสมอว่ามีซีพียูออกมาคั่นระหว่างกลางของเบอร์มาตรฐาน ซึ่งได้แก่ 8086, 80186, 80286, 80386, 80486 ตั้งแต่ 80386SX ซึ่งทำให้เราหลายคนเกิดความสับสนระหว่าง 80386 หรือ 80386DX กับ 80386SX ทางวารสารไมโครฯก็พยายามให้ข้อมูลเหล่านี้มาตลอด และวันนี้เราจะมาศึกษาหาข้อมูลของ 80486SX ซึ่งเป็นรุ่นน้องของ 80486DX แต่เป็นรุ่นพี่ของ 80386DX สาเหตุของการปรากฏตัวของ 80486SX นี้หลาย ๆ บุคคลในวงการคอมพิวเตอร์ก็เชื่อว่ามาจากการที่อินเทลแพ้คดีในการต่อสู้เรื่องลิขสิทธิ์การใช้ชื่อ 80386 กับค่าย AMD โดยตอนนี้ทางค่าย AMD ก็ได้ออก 80386 ขนาด 40 MHz ทำให้ทางบริษัทอินเทลออกซีพียูเบอร์ใหม่ในนาม 80486SX ซึ่งเมื่อพิจารณาแล้วเราจะทราบว่า 80486SX ไม่ได้แตกต่างจาก 80486DX เลย โดยเพียงนำส่วนของ math coprocessor แยกออกมาเป็น 80487SX เท่านั้น เช่นเดียวกันกับที่หลาย ๆ คนรู้ว่า 80386SX ไม่ได้แตกต่างจาก 80386DX มากมายนัก แต่ความแตกต่างระหว่าง 80386DX กับ 80486DX นั้นเรายังพอจะมองเห็นความแตกต่างได้ แม้ว่าจะไม่ชัดเจนเหมือนความแตกต่างระหว่างซีพียูเบอร์ต่าง ๆ ที่ผ่านมา (8086, 80186, 80286) ซึ่งความรู้สึกนี้ได้เกิดขึ้นกับเรา

หลาย ๆ คน ในตอนที่อินเทลประกาศตัว 80386SX ครั้งหนึ่งมาแล้ว แต่ความแตกต่างระหว่าง 80386DX กับ 80386SX ยังพอมีจุดที่เราพอมองเห็นได้ เช่น ขนาดหน่วยความจำของ 80386SX คือ 16 เมกะไบต์ ขณะที่ 80386DX ได้สูงสุด 4 กิกะไบต์ และบัสภายนอกของ 80386SX เป็นแบบ 16 บิต เท่านั้น...ฯลฯ) แม้ว่า 80486SX จะไม่มีความแตกต่างจาก 80486DX อย่างชัดเจน (ยกเว้นการแยก math coprocessor (80487SX) ออกมา) เราก็ควรรู้จักซีพียูเบอร์นี้ให้ดีมากขึ้น เพราะทางอินเทลได้หวังว่าจะใช้ซีพียูเบอร์นี้ต่อสู้กับค่าย AMD ซึ่งทำให้ราคาของ 80486 มีราคาลดลงขณะที่เราได้ประสิทธิภาพมากขึ้นประมาณ 40% เมื่อเปรียบเทียบกับ 80386 (เทียบ 80486SX ขนาด 20 MHz กับ 80386DX ขนาด 33 MHz)

คุณสมบัติของ 80486SX

- โปรแกรมระดับไบนารีสามารถใช้งานในระบบปฏิบัติการต่าง ๆ ไม่ว่าจะเป็น DOS, OS/2, windows, UNIX system V/386, IRMX, IRMK Kernel

- มีส่วนประกอบอื่นรวมอยู่ในตัว เช่น แคชขนาด 8 กิโลไบต์, Paged, Virtual Memory Management

- สามารถใช้ง่ายเพราะมีการทดสอบตัวเอง (Built in self test) มีฮาร์ดแวร์สำหรับทำดีบัก มีซอฟต์แวร์สนับสนุนจากบริษัทอินเทลหรือบริษัทอื่น ๆ

- 168 pin Grid Array, Package, 196 pin PQTP

- การออกแบบที่มีประสิทธิภาพสูงสุด คือ

1. มีส่วนแกนหลัก (core) ที่มีลักษณะเป็น 1 คำสั่งต่อ 1 สัญญาณนาฬิกา เช่นเดียวกับ 80486DX (บางคำสั่ง)

2. มีขนาดความถี่ 20 MHz (ขณะที่เขียนบทความนี้)

3. มี Burst Bus ขนาด 64 Mbyte/Sec

4. ใช้เทคโนโลยี CHMOS/V

5. มีไดนามิกบัส ขนาด 8, 16 และ 32 บิต

- มีสถาปัตยกรรมเป็นแบบ 32 บิตสมบูรณ์แบบ คือ

1. บัสของข้อมูลและบัสของแอดเดรส แบบ 32 บิต

2. รีจิสเตอร์ขนาด 32 บิต

3. มีชนิดข้อมูลแบบ 8, 16 และ 32 บิต

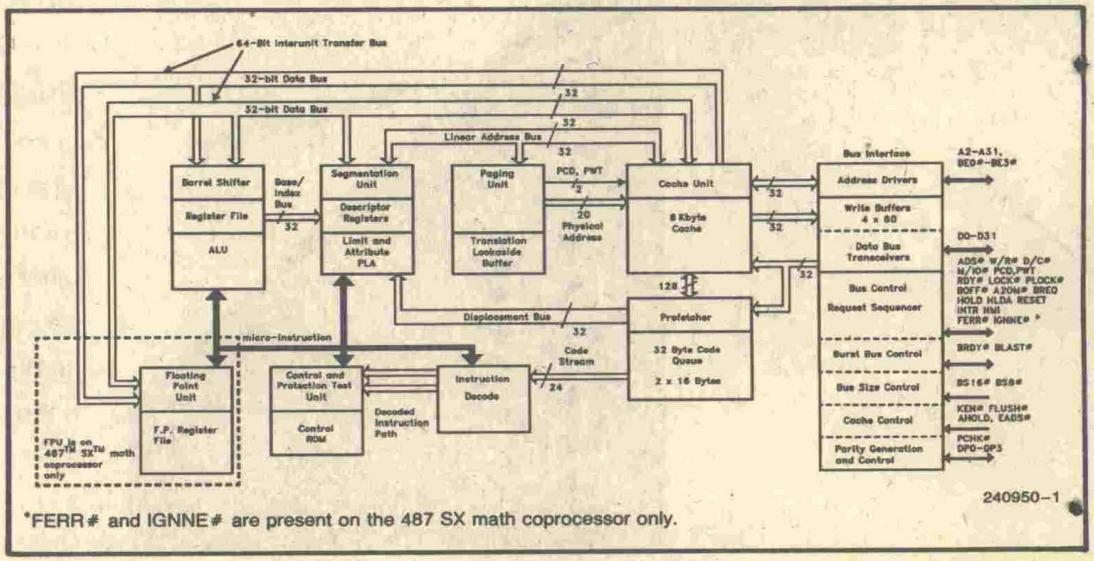
- มีการสนับสนุนการทำงานแบบหลายตัวประมวลผล (Multiprocessor support) คือ

1. มีคำสั่งสำหรับการทำงานแบบหลายตัวประมวลผล

2. มีข้อกำหนดสำหรับแคชให้มีค่าถูกต้อง

3. สนับสนุนการมีแคชตัวที่ 2 (Second Level Cache)

80486SX มีสถาปัตยกรรมแบบ 32 บิต มีส่วนจัดการหน่วยความจำ (Memory management) และหน่วยความจำแคชอยู่ภายในตัว 80486SX เอง เนื่องจาก 80486SX (+80487SX) ได้ถูกออกแบบให้มีความสามารถเท่ากับ 80486 DX จึงทำให้ 80486SX (+80487SX) สามารถใช้ซอฟต์แวร์ที่ถูกเขียนขึ้นมาสำหรับโปรเซสเซอร์ในตระกูล 80×86 ทั้งหมดได้ มีการเพิ่มความสามารถจาก 80386 ซึ่งได้แก่มีแคชขนาด 8 กิโลไบต์



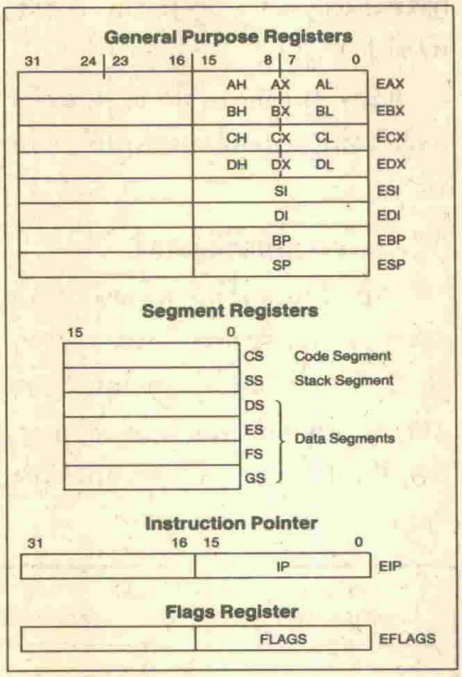
*FERR# and IGNNE# are present on the 487 SX math coprocessor only.

รูปที่ 1 โครงสร้างของ 80486SX/80487SX

แบบ 4-way set ทำให้ข้อมูลและคำสั่งที่ใช้บ่อย ๆ มีอยู่ในตัว 80486SX เอง เป็นการลดการเข้าถึง (access) ข้อมูลภายนอก มีการออกแบบโดยใช้ RISC เทคโนโลยีทำให้ลดระยะเวลาในการประมวลคำสั่งจนบางคำสั่งที่สามารถประมวลผลได้ภายในระยะเวลา 1 สัญญาณนาฬิกา มี burst bus ทำให้การดึงข้อมูลหรือคำสั่งมีความรวดเร็วขึ้นมาก คุณสมบัติและความสามารถที่เพิ่มขึ้นนี้ทำให้ 80486SX มีประสิทธิภาพเป็น 2 เท่าของ 80386 (ตามคำบอกกล่าวของอินเทล) ส่วนบริหารหน่วยความจำ (Memory Management Unit : MMU) จะประกอบด้วยส่วนการทำงาน 2 ส่วนคือ segmentation unit และ paging unit จะใช้ความสามารถของ segmentation unit อีกที่หนึ่ง ซึ่ง paging unit สามารถแบ่งเซกเมนต์ธรรมดาเป็นเซกเมนต์เดียวจนถึงมากกว่า 4 กิโลไบต์เซกเมนต์ เพื่อการทำงานแบบ virtual memory หน่วยความจำจะถูกบริหารในลักษณะเป็นเซกเมนต์ โดย 80486SX สามารถมีจำนวนเซกเมนต์สูงสุด 16,381 เซกเมนต์ แต่ละเซกเมนต์มีขนาดได้สูงสุด 4 กิกะไบต์ (2^{32} ไบต์) ซึ่งทำให้งาน (task) แต่ละงานสามารถใช้หน่วยความจำได้สูงสุด 64 เทราไบต์ เมื่อทำเซกเมนต์ให้ทำงานแบบ virtual memory สำหรับรีจิสเตอร์ของ 80486SX นั้นจะแบ่งเป็นรีจิสเตอร์พื้นฐานและรีจิสเตอร์ของระบบโดยจะกล่าวรายละเอียดต่อไป

รีจิสเตอร์พื้นฐาน

รีจิสเตอร์พื้นฐานจะเป็นไปตามรูปที่ 2 ซึ่งจะประกอบด้วยส่วนต่าง ๆ ดังนี้



รูปที่ 2 รีจิสเตอร์พื้นฐานของ 80486SX

1. รีจิสเตอร์ใช้งานทั่ว ๆ ไป

รีจิสเตอร์ใช้งานทั่ว ๆ ไปมีไว้สำหรับการทำงานของขนาด 1, 8, 6 และ 32 บิต โดยมีรีจิสเตอร์ทำงานแบบ 32 บิต คือ EAX, EBX, ECX, EDX, ESI, EDI, EBP และ ESP ซึ่งรีจิสเตอร์เหล่านี้สามารถใช้เป็นรีจิสเตอร์ 16 บิต ได้เป็น AX, BX, CX, DX, SI, DI, DP และ SP นอกจากนี้รีจิสเตอร์ 4 ตัวแรกสามารถนำมาเป็นรีจิสเตอร์ขนาด 8 บิตได้คือ AH, AL, BH, BL, CH, CL, DH และ DL ซึ่งทำให้สะดวกในการเข้าถึงข้อมูลแต่ละประเภท

2. รีจิสเตอร์สำหรับชี้คำสั่ง

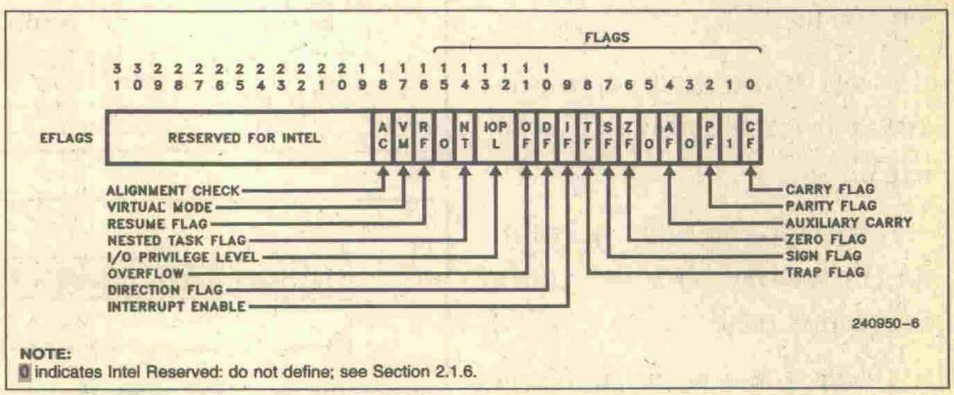
รีจิสเตอร์ที่ใช้ชี้คำสั่งก็คือ EIP ซึ่งจะมีความสัมพันธ์กับ Code segment (CS) register โดย EIP เป็นตัวชี้คำสั่งภายในเซกเมนต์ที่กำหนดโดย CS นั้นเอง รีจิสเตอร์ EIP นี้ถ้าใช้แบบ 16 บิตเราจะได้เป็น IP ซึ่งคือ lower 16 bit ของ EIP มาเป็นตัวชี้คำสั่งแทน

3. แฟลกรีจิสเตอร์

เป็นรีจิสเตอร์แสดงสถานะการทำงานของ 80486SX ดังรูปที่ 3 โดยแต่ละบิตมีความหมายคร่าว ๆ ดังนี้

3.1 AC (Alignment Check บิตที่ 18) เป็นตัวกำหนดว่าจะมีกำเนิด faults ขึ้นหรือไม่ ถ้ามีการผิดพลาดของ alignment (ดูข้อมูลได้จากตารางที่ 1) โดย AC = 1 จะหมายถึงให้เกิด faults ขึ้น (Enable)

3.2 VM (Virtual 8086 mode, บิต 17) เป็นตัวกำหนดว่าจะทำ virtual 8086 mode หรือไม่ ถ้าถูกเซต (vm = 1) ในขณะนี้ 80486SX ทำงานในโปรเทคโหมดจะทำให้ 80486SX เปลี่ยนมาทำงานเป็น Virtual 8086 mode



รูปที่ 3 แฟลกรีจิสเตอร์

3.3 RF (Resume flag, บิต 16) RF flag จะใช้ร่วมกับ debug register breakpoint

3.4 NT (Nested task บิต 14) เป็นตัวบอกว่างาน (task) ที่กำลังทำนี้เป็นส่วนหนึ่งของงาน (task) อื่นหรือไม่

3.5 IOPL (Input/Output Privilege Leved, บิต 12-13) เป็นตัวกำหนดค่าสูงสุดของ CPL (Current Privilege Level) มีอทำคำสั่งเกี่ยวกับ I/O โดยเกิด exception 13

3.6 OF (Overflow flag บิต 11) เป็นตัวบอกว่าผลของการประมวลผลมีการล้นเกิน (overflow) หรือเปล่า

3.7 DF (Direction flag บิต 10) เป็นตัวกำหนดทิศทาง (ลดค่าหรือเพิ่มค่า ESI และ EDI) ในการทำคำสั่งเกี่ยวกับสตริง

3.8 IF (Inter Enable Flag, บิต 9) เป็นตัวกำหนดว่าจะให้มีการตอบสนองการอินเตอร์รัพต์หรือไม่

3.9 TF (Trap Enable flag บิต 8) เป็นตัวกำหนดการเกิด exception 1 ในการทำ single-step

3.10 SF (Sign flag บิต 7) บิตนี้แสดงเครื่องหมายของการคำนวณซึ่งก็คือ ค่าของบิตสูงสุดของการคำนวณแบบ 8, 16 และ 32 บิต

3.11 ZF (Zero flag บิต 6) เป็นบิตแสดงผลการประมวลผลว่าเป็นศูนย์หรือไม่

3.12 AF (Auxiliary flag บิต 4) บิตช่วยทด (มีการทดระหว่าง 4 บิต) ในการคำนวณแบบ BCD

3.13 PF (Parity flag บิต 2) เป็นตัวบอกว่าจำนวนเลข 1 ใน 8 บิตล่างของการกระทำต่าง ๆ มีจำนวนเป็นเลขคู่หรือไม่

3.14 CF (Carry flag บิต 0) บอกถึงการบวกมีการทดและการลบมียืมหรือไม่

4. เซกเมนต์รีจิสเตอร์

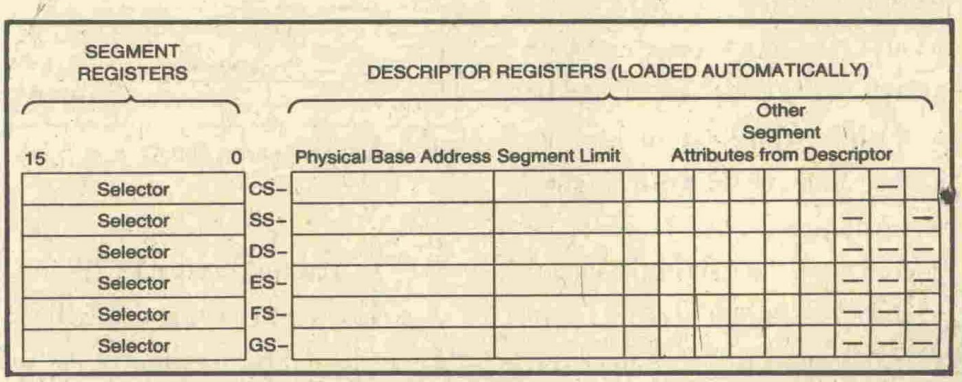
เป็นรีจิสเตอร์ที่กำหนดเซกเมนต์ต่าง ๆ ในการทำงานของโปรแกรมต่างๆ โดยมี CS เป็นตัวระบุเซกเมนต์ของคำสั่ง SS เป็นตัวกำหนด stack segment, DE, ES, FS, GS แสดงเซกเมนต์ของข้อมูล

5. Segment Descriptor Cache Registers

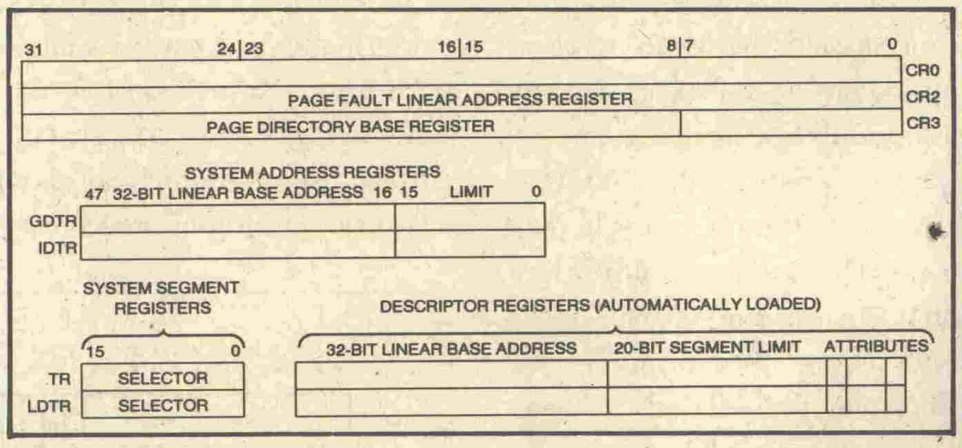
เป็นตัวบอกลักษณะของเซกเมนต์รีจิสเตอร์ซึ่งมองไม่เห็นค่าของรีจิสเตอร์เหล่านี้โดยโปรแกรมเมอร์ ซึ่งรีจิสเตอร์เหล่านี้จะถูกเปลี่ยนค่า (update) ให้ถูกต้องอย่างอัตโนมัติ

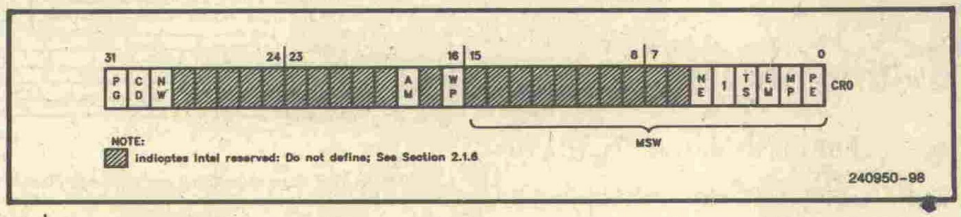
รีจิสเตอร์ของระบบ (system level rigisters)

เป็นรีจิสเตอร์สำหรับบอกสถานะและควบคุมการทำงานของ 80486SX ซึ่งประกอบด้วยรีจิสเตอร์ต่าง ๆ ดังนี้



รูปที่ 4 เซกเมนต์รีจิสเตอร์และ Segment Descriptor Cach Registers





รูปที่ 5 Control Register 0

1. Control Register

ประกอบด้วยรีจิสเตอร์ 4 ตัวคือ

1.1 Control Register (CR) มีบิตเพิ่มขึ้นคือ CD, NW, AM, WP และ NE ซึ่งแต่ละบิตของ CR มีความหมายดังนี้

- PG (Paging Enable, บิต 31) บอกถึงการทำงานแบบเพจจิ้ง ว่าจะให้ทำงานแบบเพจจิ้งหรือไม่

- CD (Cache Disable, บิต 30) เป็นตัวบอกว่าจะใช้แคชภายในหรือไม่

- NW (Not write through บิต 29) เป็นตัวกำหนดการทำ write throughs และ write invalidate cycles ของแคช

- AM (Alignment Mask, บิต 18) เป็นตัวกำหนดการใช้บิต AC ของ Eflag ว่าจะตรวจเช็คบิต AC หรือไม่ (ทำ alignment)

- WP (Write protect, บิต 16) ใน 80386 สามารถกำหนดการอ่านอย่างเดียว (Read Only) ใน privilege Level 0-2 ซึ่งจะตรงกับ WP=0 แต่ถ้า WP=1 เราจะกำหนดได้ทุก Level

- NE (Numerics Exception, บิต 5) เป็นตัวควบคุมการเกิด interrupt 7

- PE (Protection Enable) เป็นตัวกำหนดการทำงานระบบโปรเทคโหมด

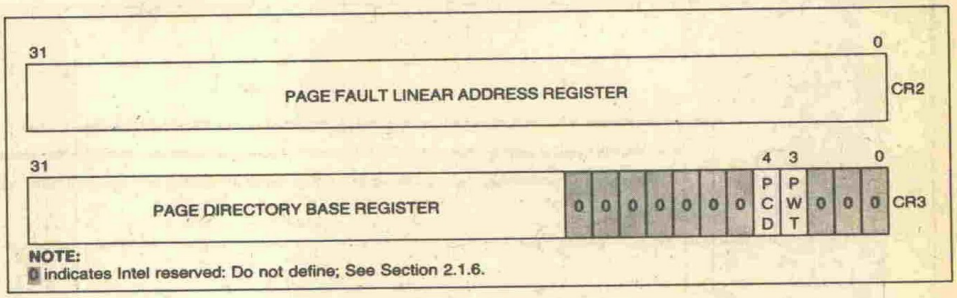
- EM (Emulate Coprocessor บิต 2) เป็นตัวกำหนดการทำงานของคำสั่งเกี่ยวกับ floating point ว่าถ้าเกิด interrupt 7 จะให้วิ่งมาที่ซอฟต์แวร์ [ถ้า Em=0 จะทำให้ระบบหยุดทำงาน (system hang)]

- MP (Monitor Coprocessor บิต 1) สำหรับการทำงานปกติของ 80486SX ค่าของ MP จะเท่ากับ 0 (TS บิตจะไม่ได้รับการสนใจ) แต่ถ้า MP=1 จะใช้งานร่วมกับ TS บิต เพื่อกำหนดการทำงานของคำสั่ง WAIT

- TS (Task Switch บิต 3) ใช้กำหนดการเปลี่ยนงาน (task)

1.2 Control Register 1 (CR1) สงวนไว้ใช้งานในภายหน้า

1.3 Control Register 2 และ 3 (CR2, CR3) ใช้กำหนดการทำงานในแบบเพจจิ้ง



รูปที่ 6 Control Register 2 และ 3

2. System address Registers

เป็นตัวกำหนดการใช้หน่วยความจำส่วนต่าง ๆ ของระบบซึ่งประกอบด้วยรีจิสเตอร์ 4 ตัวคือ GDT (Global Descriptor Table) ใช้ระบุการใช้งานของหน่วยความจำทั้งหมด, LDT (Local Descriptor Table) ใช้ระบุการใช้งานของหน่วยความจำในแต่ละงาน (task), IDT (Interrupt Descriptor Table) ใช้ระบุตำแหน่งของตารางอินเตอร์รัพต์, TSS (Task State Segment) ใช้ระบุคุณสมบัติและสถานะของงานแต่ละงาน

Debug Registers

LINEAR BREAKPOINT ADDRESS 0	DR0
LINEAR BREAKPOINT ADDRESS 1	DR1
LINEAR BREAKPOINT ADDRESS 2	DR2
LINEAR BREAKPOINT ADDRESS 3	DR3
Intel Reserved Do Not Define	DR4
Intel Reserved Do Not Define	DR5
BREAKPOINT STATUS	DR6
BREAKPOINT CONTROL	DR7

Test Registers

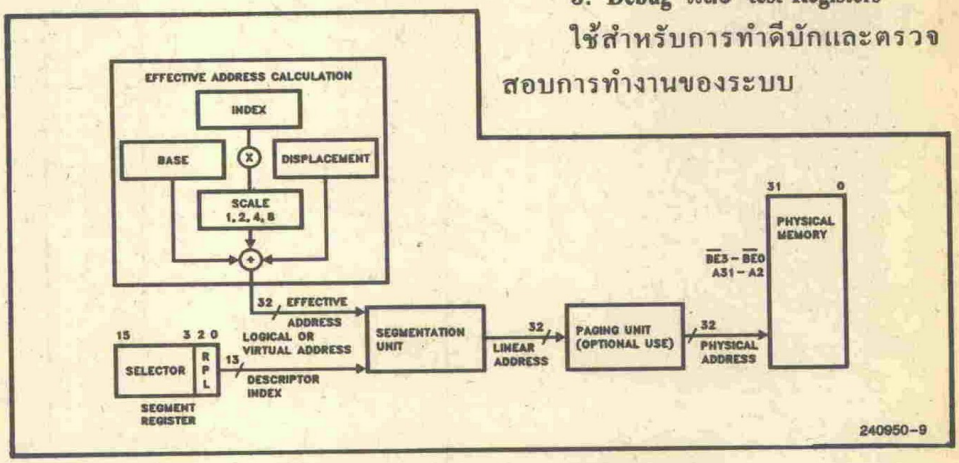
CACHE TEST DATA	TR3
CACHE TEST STATUS	TR4
CACHE TEST CONTROL	TR5
TLB TEST CONTROL	TR6
TLB TEST STATUS	TR7

TLB = Translation Lookaside Buffer

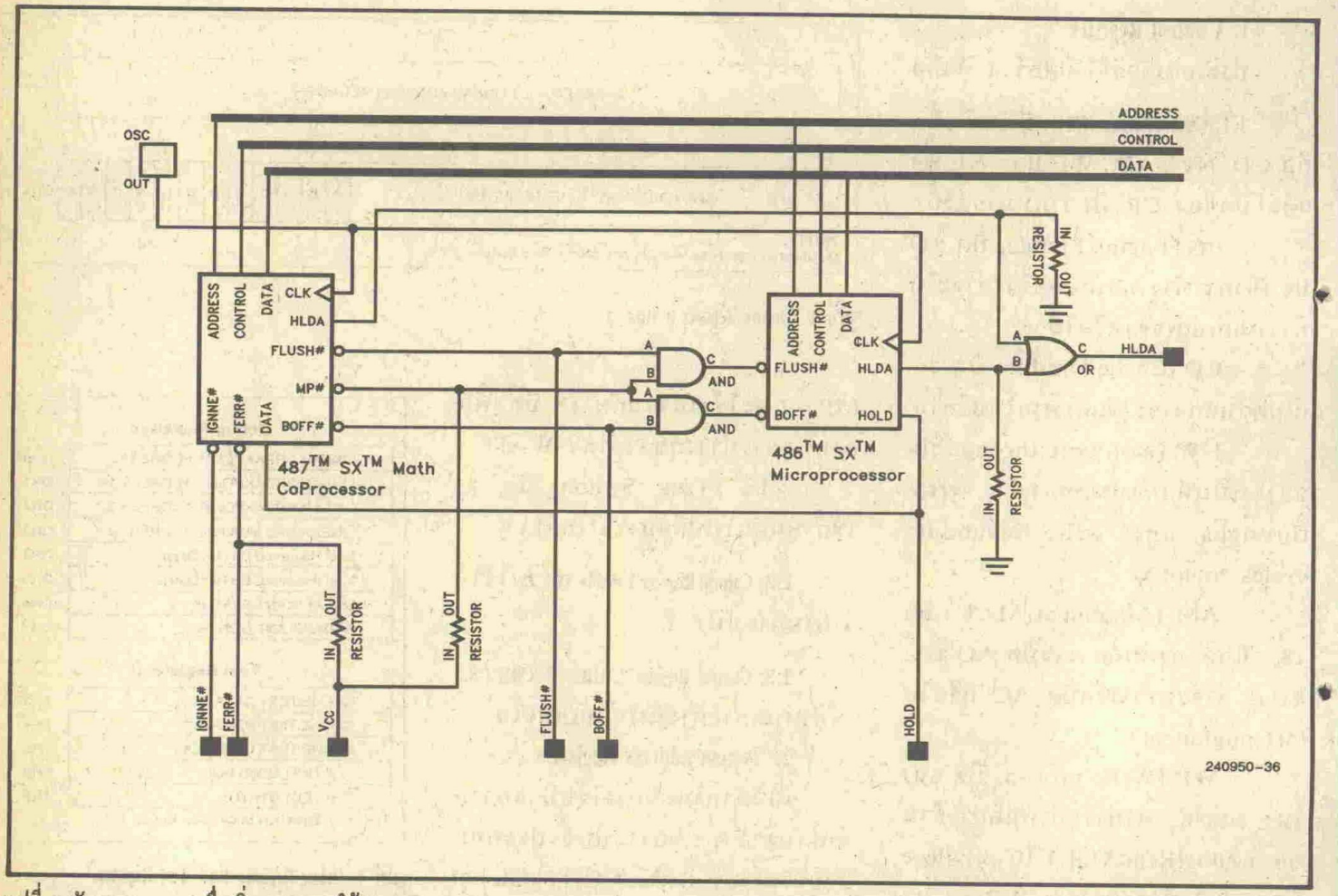
รูปที่ 7 Debug Register และ Test Registers

3. Debug และ test Registers

ใช้สำหรับการทำดีบักและตรวจสอบการทำงานของระบบ

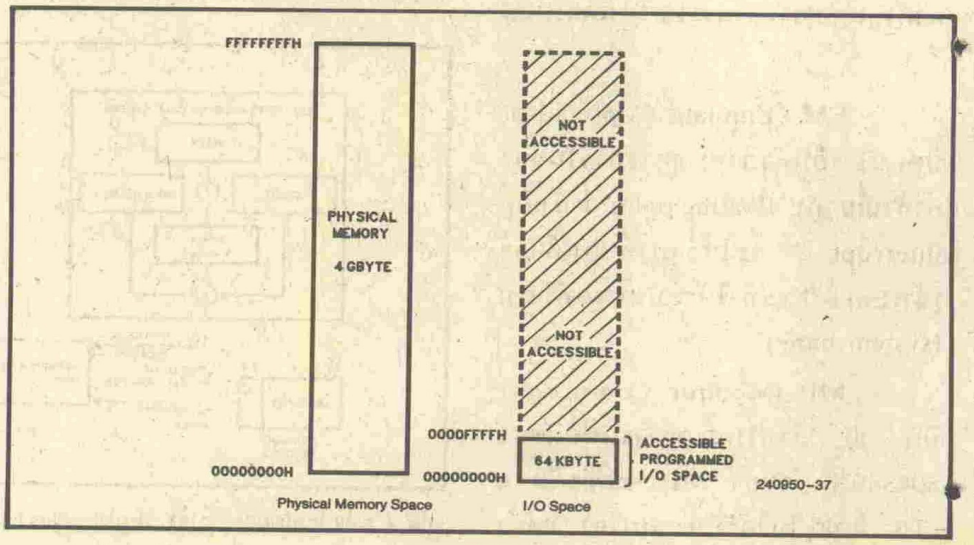


รูปที่ 8 การแปลงแอดเดรสอ้างอิง (logical address) ไปเป็นแอดเดรสจริง (physical address)

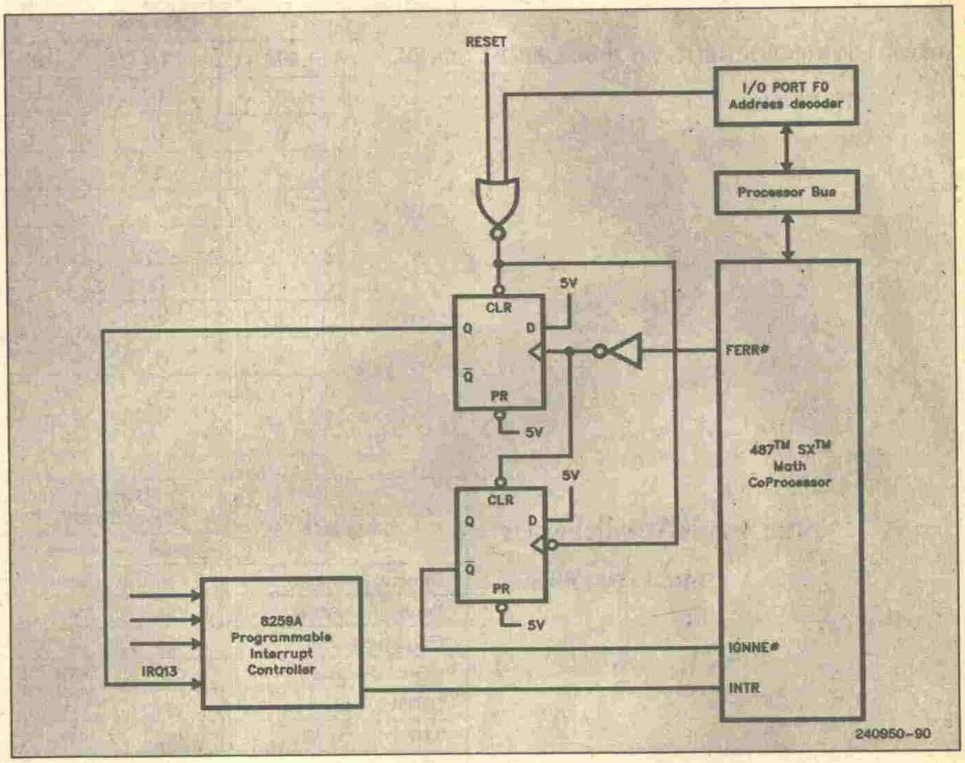


รูปที่ 9 ลักษณะของวงจรเพื่อเพิ่ม 80487SX ให้ระบบ

รูปที่ 10 Physical Memory และ I/O spaces



รูปที่ 11 DOS Compatible Numerices Error Curcuit



ตารางที่ 1 ชนิดของข้อมูลที่ต้องการทำ alignment

Memory Access	Alignment (Byte Boundary)
Word	2
Dword	4
Single Precision Real	4
Double Precision Real	8
Extended Precision Real	8
Selector	2
48-Bit Segmented Pointer	4
32-Bit Flat Pointer	4
32-Bit Segmented Pointer	2
48-Bit "Pseudo-Descriptor"	4
FSTENV/FLDENV Save Area	4/2 (On Operand Size)
FSAVE/FRSTOR Save Area	4/2 (On Operand Size)
Bit String	4

ตารางที่ 2 โหมดการทำงานของ co-processor

PG	PE	Mode
0	0	REAL Mode. Exact 8086 semantics, with 32-bit extensions available with prefixes.
0	1	Protected Mode. Exact 80286 semantics, plus 32-bit extensions through both prefixes and "default" prefix setting associated with code segment descriptors. Also, a sub-mode is defined to support a virtual 8086 within the context of the extended 80286 protection model.
1	0	UNDEFINED. Loading CR0 with this combination of PG and PE bits will raise a GP fault with error code 0.
1	1	Paged Protected Mode. All the facilities of Protected mode, with paging enabled underneath segmentation.

ตารางที่ 3 โหมดการทำงานของ On-clip Cache

CD	NW	Operating Mode
1	1	Cache fills disabled, write-through and invalidates disabled.
1	0	Cache fills disabled, write-through and invalidates enabled.
0	1	INVALID. If CR0 is loaded with this configuration of bits, a GP fault with error code is raised.
0	0	Cache fills enabled, write-through and invalidates enabled.

ตารางที่ 4 แสดงสถานะการทำงานของ 80486SX/80487SX ตาม EM, TS, MP บิต

CR0 Bit			Instruction Type	
EM	TS	MP	Floating Point	Wait
0	0	0	Execute	Execute
0	0	1	Execute	Execute
0	1	0	Exception 7	Execute
0	1	1	Exception 7	Exception 7
1	0	0	Exception 7	Execute
1	0	1	Exception 7	Execute
1	1	0	Exception 7	Execute
1	1	1	Exception 7	Exception 7

ตารางที่ 5 การใช้รีจิสเตอร์ในโหมดการทำงานต่าง ๆ ของ 80486SX

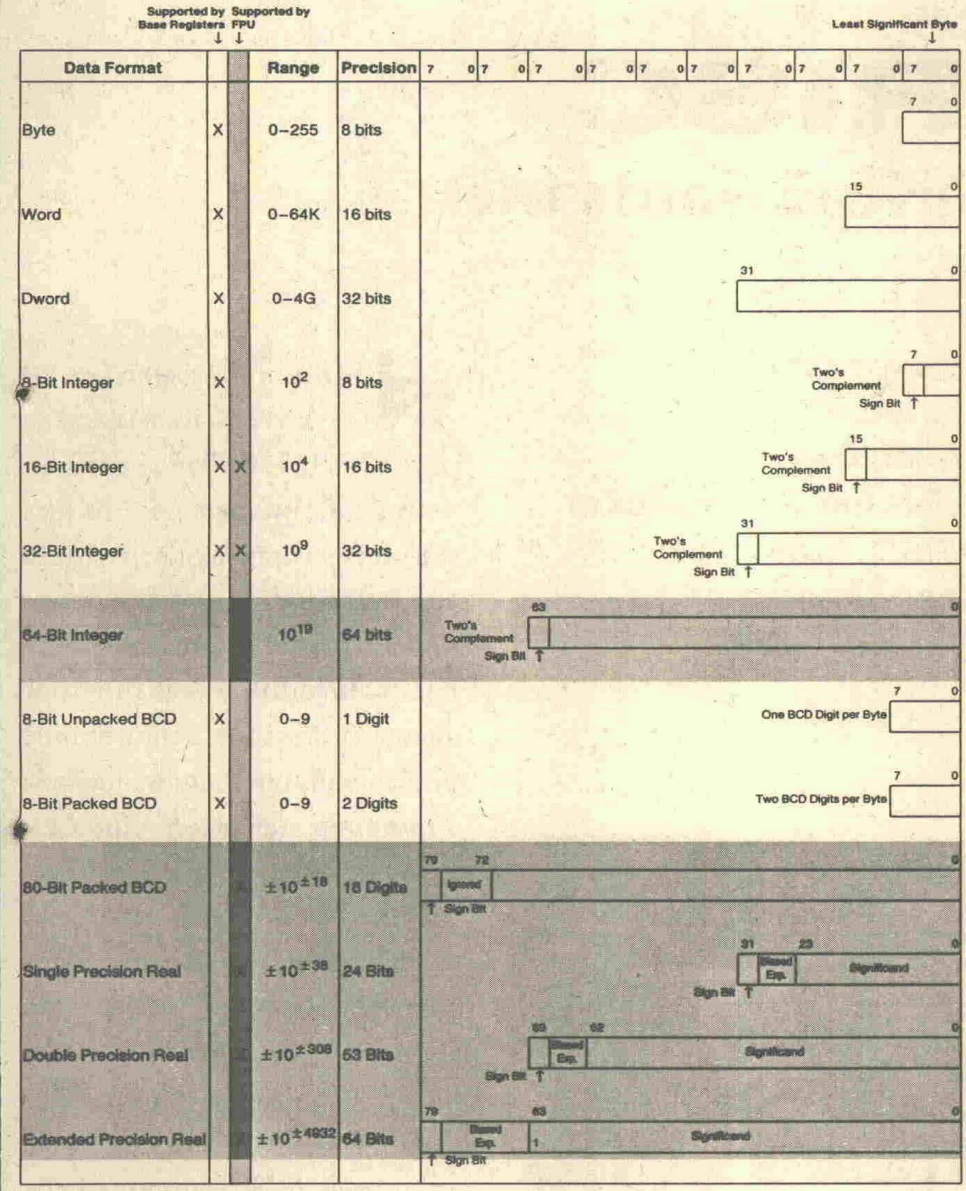
Register	Use in Real Mode		Use in Protected Mode		Use in Virtual 8086 Mode	
	Load	Store	Load	Store	Load	Store
General Registers	Yes	Yes	Yes	Yes	Yes	Yes
Segment Register	Yes	Yes	Yes	Yes	Yes	Yes
Flag Register	Yes	Yes	Yes	Yes	IOPL	IOPL*
Control Registers	Yes	Yes	PL = 0	PL = 0	No	Yes
GDTR	Yes	Yes	PL = 0	Yes	No	Yes
IDTR	Yes	Yes	PL = 0	Yes	No	Yes
LDTR	No	No	PL = 0	Yes	No	No
TR	No	No	PL = 0	Yes	No	No
FPU Data Registers	Yes	Yes	Yes	Yes	Yes	Yes
FPU Control Registers	Yes	Yes	Yes	Yes	Yes	Yes
FPU Status Registers	Yes	Yes	Yes	Yes	Yes	Yes
FPU Instruction Pointer	Yes	Yes	Yes	Yes	Yes	Yes
FPU Data Pointer	Yes	Yes	Yes	Yes	Yes	Yes
Debug Registers	Yes	Yes	PL = 0	PL = 0	No	No
Test Registers	Yes	Yes	PL = 0	PL = 0	No	No

ตารางที่ 6 กฎการเลือกใช้เซกเมนต์รีจิสเตอร์

Type of Memory Reference	Implied (Default) Segment Use	Segment Override Prefixes Possible
Code Fetch	CS	None
Destination of PUSH, PUSHF, INT, CALL, PUSHA Instructions	SS	None
Source of POP, POPA, POPF, IRET, RET instructions	SS	None
Destination of STOS, MOVS, REP STOS, REP MOVS Instructions (DI is Base Register)	ES	None
Other Data References, with Effective Address Using Base Register of:		
[EAX]	DS	All
[EBX]	DS	
[ECX]	DS	
[EDX]	DS	
[ESI]	DS	
[EDI]	DS	
[EBP]	SS	
[ESP]	SS	

NEW TECHNOLOGY

ก้าวทันสมัยด้วยการรู้จัก 80486SX



Data Format			Range	Precision
Byte	X		0–255	8 bits
Word	X		0–64K	16 bits
Dword	X		0–4G	32 bits
8-Bit Integer	X		10^2	8 bits
16-Bit Integer	X	X	10^4	16 bits
32-Bit Integer	X	X	10^9	32 bits
64-Bit Integer			10^{19}	64 bits
8-Bit Unpacked BCD	X		0–9	1 Digit
8-Bit Packed BCD	X		0–9	2 Digits
80-Bit Packed BCD		X	$\pm 10^{\pm 18}$	18 Digits
Single Precision Real		X	$\pm 10^{\pm 38}$	24 Bits
Double Precision Real		X	$\pm 10^{\pm 308}$	53 Bits
Extended Precision Real		X	$\pm 10^{\pm 4932}$	64 Bits

ตารางที่ 7 ชนิดของข้อมูลที่สนับสนุนโดย 80486SX/80487SX

Register	Initial Value (BIST)	Initial Value (No Bist)
EAX	Zero (Pass)	Undefined
ECX	Undefined	Undefined
EDX	0400 + Revision ID	0400 + Revision ID
EBX	Undefined	Undefined
ESP	Undefined	Undefined
EBP	Undefined	Undefined
ESI	Undefined	Undefined
EDI	Undefined	Undefined
EFLAGS	00000002h	00000002h
EIP	0FFF0h	0FFF0h
ES	0000h	0000h
CS	F000h*	F000h*
SS	0000h	0000h
DS	0000h	0000h
FS	0000h	0000h
GS	0000h	0000h
IDTR	Base = 0, Limit = 3FFh	Base = 0, Limit = 3FFh
CR0	60000010h	60000010h
DR7	00000000h	00000000h
CW	037Fh	Unchanged
SW	0000h	Unchanged
TW	FFFFh	Unchanged
FIP	00000000h	Unchanged
FEA	00000000h	Unchanged
FCS	0000h	Unchanged
FDS	0000h	Unchanged
FOP	000h	Unchanged
FSTACK	Undefined	Unchanged

ตารางที่ 8 ค่าของรีจิสเตอร์ต่าง ๆ ภายในจากการทำรีเซต

สำหรับบทความตอนนี้เราจะรู้จัก 80484SX เพียงเท่านี้ โดยเราทำความรู้จักรีจิสเตอร์ต่าง ๆ ที่กล่าว ส่วนการแปลแอดเดรส, memory/IO space, ชนิดข้อมูล สามารถดูได้จากรูปและตารางต่าง ๆ ที่ผู้เขียนนำมาฝากและถ้าท่านผู้อ่านต้องการรายละเอียดมากกว่านี้ สามารถศึกษาได้จากคู่มือหรือบอกกล่าวมาทางไมโครแล้วเราจะพบกันอีก สวัสดีครับ